PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-134457
(43)Date of publication of application : 18.05.2001

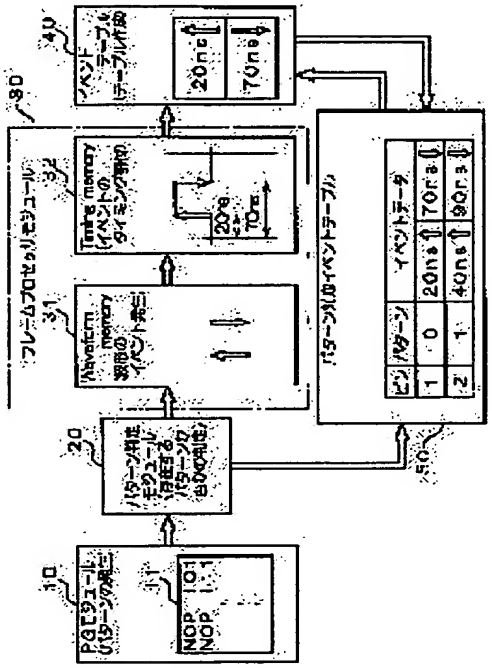
(51)Int.CI. G06F 11/22
G06F 11/28
// G01R 31/28

(21)Application number : 11-316160
(22)Date of filing : 08.11.1999

(71)Applicant : ADVANTEST CORP
(72)Inventor : AZUMA SHINSAKU
FUKUSHIMA KIYOSHI

(54) PROGRAM DEBUG DEVICE FOE SEMICONDUCTOR TEST

(57)Abstract:
PROBLEM TO BE SOLVED: To quickly generate waveform data corresponding to each pin even if emulating the operation of a semiconductor device.
SOLUTION: Event data generated by a frame processor module 30 are stored with pattern data in a pattern corresponding event table 50. Whether or not the pattern data outputted from a pattern generating module 10 are present in the pattern corresponding event table 50 is decided by a pattern judging module 20. As the result of decision, when the pattern data are already present, the event data corresponding to the pattern data are read from the pattern corresponding event table 50, and stored in an event table 40. When any pattern data are not present, the normal event data generation processing is operated by the frame processor module 30.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]
[Claim 1] A pattern generating means to output the data of two or more bit configuration based on a pattern program, Based on the data of a bit pattern, two or more said data about a wave-like event are generated from wave memory. A frame processor means to output the event data which generate the timing data in which the generating timing of an event is shown from timing memory, and consist of data about said event, and said timing data, An event table means to store said event data outputted from said frame processor means, Said event table means corresponding to the pattern which relates with the data of a bit pattern, and is stored which became the origin of the event data's generating of said event data stored in said event table means, [two or more] When [said / by which two or more data of a bit pattern exist in said event table corresponding to a pattern] outputted from said pattern generating means, said said event data corresponding to [two or more] the data of a bit pattern are read from said event table means corresponding to a pattern. Program debugging equipment for a semi-conductor trial characterized by being constituted including said pattern judging means to output two or more data of a bit pattern to said frame processor means when it does not store and exist in said event table means.
[Claim 2] Program debugging equipment for a semi-conductor trial characterized by said thing [that two or more said event data are stored by making the data of a bit pattern into the address] in claim 1 at said event table means corresponding to a pattern.

[Translation done.]

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Field of the Invention] This invention relates to the program debugging equipment for a semi-conductor trial which emulates actuation of a semi-conductor testing device and performs the program verification for a trial.
[0002]
[Description of the Prior Art] From the former, the semi-conductor testing device is known as what performs a direct-current trial, a functional test, etc. to various kinds of semiconductor devices, such as a logic IC before shipment, and semiconductor memory. The trial which a semi-conductor testing device performs is divided roughly, and are a functional test and a direct-current trial. A functional test gives a predetermined test pattern signal to the semiconductor device for inspected, and inspects whether the semiconductor device for inspected performed prearranged actuation to this test pattern signal. A direct-current trial inspects whether the property which the direct-current property of each terminal of the semiconductor device for inspected planned is fulfilled. For example, when the voltage-source-current-measurement trial which examines whether a prearranged current can take out from a terminal when a known electrical potential difference is impressed, or a known current is passed or taken out, there is a current-source-voltage-measurement trial which examines whether the prearranged electrical potential difference has occurred for the terminal. Moreover, even when performing a functional test, set the electrical potential difference at the time of high level as 4 volts of the electrical-potential-difference value of normal, for example, a value lower than 5 volts, the electrical potential difference at the time of a low level is set as 0.5 volts of the electrical-potential-difference value of normal, for example, a value higher than 0 volt, or it carries out in many cases by changing various electrical-potential-difference conditions impressed to the semiconductor device for inspected, current conditions, etc.
[0003] Since various kinds of conditions of what kind of item to examine on what kind of conditions are beforehand included in the program for a semi-conductor trial when performing a functional test and a direct-current trial, the various trials of the semiconductor device for inspected can be performed by operating this program for a semi-conductor trial. However, the program for a semi-conductor trial must control the actuation across which it goes variably, such as a setup of a trial item, a setup of a test condition, experimental activation, and a judgment of a test result, and is built by the program of a huge step. When the class of semiconductor device for inspected is changed or that logic is changed, this program for a semi-conductor trial is combined with it, and must be changed variously. When the program for a semi-conductor trial is created newly or is changed, the program itself must evaluate the program for whether it is what operates normally.
[0004] To the semiconductor device for inspected which the quality understands beforehand using the semi-conductor testing device actual as law on the other hand, the program for a semi-conductor trial was operated and the program was evaluated. However, the semi-conductor testing device itself is expensive, evaluating whether the program for a semi-conductor trial operates normally from there being little introductory number using an actual

semi-conductor testing device will stop Rhine of a semi-conductor trial, and it is not desirable. Then, it was verifying whether conventionally, the program for a semi-conductor trial would not be evaluated using an actual semi-conductor testing device, but actuation of a semi-conductor testing device would be emulated using general purpose computers, such as a workstation, and the program for a semi-conductor trial would operate normally.

[0005] Thus, there is a thing which was indicated by JP,9-185519,A to emulate actuation of a semi-conductor testing device. This is related with the debugging equipment for examining whether the program for a semi-conductor trial operates normally. This debugging equipment constitutes the false semi-conductor testing device by operating the program for a semi-conductor trial which serves as a candidate for debugging under the operating system of a general purpose computer. An actual semi-conductor testing device this debugging equipment the same with examining to the actual semiconductor device for inspected The impression data point which followed the test condition to each pin (pin for evaluation) used as the measuring object of the semiconductor device for inspected is created. By creating in false the output data point which will be outputted from the output pin, when this impression data point is impressed to the input pin of the semiconductor device for inspected, and comparing this output wave with a test condition Pass/fail was judged, it was stored in the test-result storing section, comparison examination of the expected value of the test result expected to be it was carried out, and it was verifying whether the program for a semi-conductor trial would operate normally.

[0006] Drawing 4 is drawing showing the outline configuration of the driver wave generating section which creates the impression data point according to a test condition in conventional debugging equipment. In this drawing, the driver wave generating section is constituted including the pattern generating module (PG module) 10, the frame processor module 30, and the event table 40. The pattern generating module 10 generates the pattern data of two or more bit configuration according to the pattern program 11, and outputs them to the frame processor module 30. For example, the pin data which specify the pin by which a wave is impressed to the data of 5 bit patterns for which it opts with the combination of the data of the triplet configuration which the usual pattern generator generates, the timing set of 4 bit patterns for choosing 16 kinds of timing edge, and the mode data about pattern generating mode were added, and the pattern data of two or more bit configuration outputted from the pattern generating module 10 are constituted. Pin data are added because the classes (an NRZ wave, NZ wave, an SBC wave, FIX wave, etc.) of impression wave can be set up now for every pin.

[0007] The frame processor module 30 is constituted including the wave memory (Waveform memory) 31 and the timing memory (Timing memory) 32, generates the event data based on the pattern data outputted from the pattern generating module 10, and outputs them to the event table 40. The wave memory 31 generates the data about a wave-like event based on the pattern data outputted from the pattern generating module 10. The data about this wave-like event are data in which the data in which a wave-like standup is shown, or wave-like falling is shown. The timing memory 32 assigns the timing data in which the generating timing of the event is shown to the data about the wave-like event outputted from the wave memory 31 based on the pattern data outputted from the pattern generating module 10, generates event data (data which consist of combination of the data and timing data about an event), and outputs it to the event table 40. Sequential storing of the event data outputted to the event table 40 from the frame processor module 30 is carried out as a data point corresponding to each pin.

[0008] It explains that the driver wave generating section of drawing 4 operates how according to the pattern program 11. In addition, the pattern data of two or more bit configuration outputted from the pattern generating module 10 shall consist of the impression pattern data and pin data within the pattern program 11, and are omitted and explained about a timing set and mode data. First, according to the 1st NOP instruction "NOP !01", the pattern generating module 10 outputs the pattern data "10" corresponding to the pin data "1" and impression pattern data "0" of the 1st pin to the frame processor module 30. The wave memory 31 generates two, the event which shows a wave-like standup based on pattern data "10", and the event which shows falling. The timing memory 32 assigns 70ns as generating timing of an event which shows falling for 20ns as generating timing of an event which shows a standup.

Consequently, the data point which consists of event data of starting in timing 20ns, and event data of falling in timing 70ns is stored in the event table 40.
[0009] Next, the pattern generating module 10 outputs the pattern data "21" corresponding to the pin data "2" and impression pattern data "1" of the 2nd pin of the 1st NOP instruction "NOP !01" to the frame processor module 30. The frame processor module 30 generates the event data based on pattern data "21", and stores them in the event table 40 by making it into a data point. Hereafter, similarly, the processing based on the pattern data "11" corresponding to the pin data "1" and impression pattern data "1" of the 1st pin of the 2nd NOP instruction "NOP !11" and the pattern data "21" corresponding to the pin data "2" and impression pattern data "1" of the 2nd pin is repeated successively, and actuation of the driver wave generating section of a semi-conductor testing device is emulated.
[0010]
[Problem(s) to be Solved by the Invention] By the way, the actual semi-conductor testing device has the frame processor for every pin. Each frame processor is processing in juxtaposition based on the pattern data from the pattern generating section. However, conventional debugging equipment follows a pattern program using the frame processor module 30, as it is shown in drawing 4 , since he is trying to make processing that it is equivalent to an actual frame processor under the operating system of a general purpose computer perform, and it is the 1st pin and the 2nd pin. -- Generation processing of a data point is performed to the serial in sequence. That is, juxtaposition-processing which the frame processor of an actual semi-conductor testing device was performing is changed and performed to serial-processing. Therefore, it came to spend great time amount on creation processing of a data point, and had become a problem as the number of pins of the semiconductor device for inspected increased. Moreover, recently, the number of pins of the semiconductor device for inspected is in a way of an increment, and it had become an important technical problem to shorten the time amount which creation processing of a data point takes.
[0011] This invention is created in view of such a point, and that purpose is in offering the program debugging equipment for a semi-conductor trial which can create the data point corresponding to each pin at a high speed, even when actuation of a semi-conductor testing device is emulated.
[0012]
[Means for Solving the Problem] In order to solve an above-mentioned technical problem, the program debugging equipment for a semi-conductor trial indicated by claim 1 A pattern generating means to output the data of two or more bit configuration based on a pattern program, Based on the data of a bit pattern, two or more said data about a wave-like event are generated from wave memory. A frame processor means to output the event data which generate the timing data in which the generating timing of an event is shown from timing memory, and consist of data about said event, and said timing data, An event table means to store said event data outputted from said frame processor means, Said event table means corresponding to the pattern which relates with the data of a bit pattern, and is stored which became the origin of the event data's generating of said event data stored in said event table means, [two or more] When [said / by which two or more data of a bit pattern exist in said event table corresponding to a pattern] outputted from said pattern generating means, said said event data corresponding to [two or more] the data of a bit pattern are read from said event table means corresponding to a pattern. When it does not store and exist in said event table means, it is constituted including said pattern judging means to output two or more data of a bit pattern to said frame processor means.
[0013] Invention indicated by claim 1 stores in the event table means corresponding to a pattern the event data which the frame processor means generated with the data of two or more bit configuration. It judges whether the data of two or more bit configuration outputted from the pattern generating means exist in the event table means corresponding to a pattern with a pattern judging means. When it already exists, it stores in an event table means by reading the event data corresponding to the data of two or more bit configuration from there, and in not existing, it is made to perform generating processing of the usual event data based on a frame

processor means. By this, about the once generated event data, the generating processing of event data which the frame processor means was performing can be omitted, and the data point corresponding to each pin can be created now at a high speed. In addition, when the data of two or more bit configuration are repeatedly generated based on a pattern program like a semi-conductor testing device, the rate that generating processing of the event data based on a frame processor means is omitted also becomes high, and the effectiveness is very large.
[0014] The program debugging equipment for a semi-conductor trial of this invention indicated by claim 2 stores in said event table means corresponding to a pattern said said event data which make the address two or more data of a bit pattern. Since invention indicated by claim 2 can read said event data to the event table means corresponding to a pattern only by supplying the data of two or more bit configuration as the address, it can create a data point at a high speed.
[0015]
[Embodiment of the Invention] Hereafter, the gestalt of 1 operation of the program debugging equipment for a semi-conductor trial concerning this invention is explained, referring to a drawing. Drawing 2 is drawing showing the whole program debugging equipment configuration for a semi-conductor trial. By emulating actuation of a semi-conductor testing device, and simulating actuation of the semiconductor device for inspected, debugging equipment 100 is for verifying whether the program for a semi-conductor trial operates normally, and is realized by general purpose computers, such as a workstation.
[0016] Since the debugging equipment 100 concerning the gestalt of this operation simulates actuation of an actual semi-conductor testing device and the semiconductor device for inspected, before it gives that detailed explanation, it explains the outline configuration of the semi-conductor testing device simulated.
[0017] Drawing 3 is drawing showing the actual semi-conductor testing-device whole configuration. The condition that the actual semiconductor device 250 for inspected was connected to the semi-conductor testing device 200 is shown by this drawing. The semi-conductor testing device 200 performs various kinds of direct-current trials (DC parametric trial) and functional tests to the semiconductor device 250 for inspected. The semi-conductor testing device 200 is constituted including the socket section (not shown) which carries the circuit tester control section 210, the circuit tester bus 230, the circuit tester body 240, and the semiconductor device 250 for inspected.
[0018] The circuit tester control section 210 is for controlling actuation of the circuit tester body 240, and is constituted including the program 212 for a semi-conductor trial (device test program), an application program 214, the language analysis activation section 216, the circuit tester library 218, and the circuit tester bus driver 220.
[0019] The device test program 212 describes the procedure and approach for what kind of trial a user performs to the semiconductor device 250 for inspected using the semi-conductor testing device 200. Generally development creation of this device test program is done by the user of the semi-conductor testing device 200. Therefore, without using the actual semi-conductor testing device 200, a user can verify whether the device test program 212 which he created using the debugging equipment 100 concerning the gestalt of this operation operates normally, and can create a highly complete device test program. The language analysis activation section 216 performs syntax analysis of the device test program 212 etc., and plays the central role which operates the semi-conductor testing device 200 faithfully according to the device test program 212. An application program 214 cooperates with the device test program 212 and the language analysis activation section 216, operates, impresses the actual stimulus corresponding to a functional test and a direct-current trial etc. to the semiconductor device 250 for inspected, incorporates the output signal, and the quality of the semiconductor device 250 for inspected is judged, or it analyzes a property. The circuit tester library 218 directs measurement actuation to the circuit tester body 240 while it changes the instruction of the device test program 212 after syntax analysis was performed by the language analysis activation section 216 into the instruction (data about the data write-in instruction to the register 242 mentioned later, and the data read-out instruction from a register 242) of register level and

performs creation and a setup of data required for actuation of the semi-conductor testing device 200. The circuit tester bus driver 220 transmits the data created by the circuit tester library 218 to the register 242 within the circuit tester body 240 through the circuit tester bus 230.

[0020] The circuit tester body 240 performs various kinds of trials to the semiconductor device 250 for inspected based on the data from the circuit tester control section 210 incorporated through the circuit tester bus 230. The circuit tester body 240 is constituted including a register 242, memory 244, and the test activation section 246. A register 242 stores the data from the circuit tester library 218 incorporated through the circuit tester bus 230. The data stored in this register 242 are outputted to the test activation section 246 through direct or memory 244. Moreover, a register 242 and memory 244 have the test-result storing field (not shown) which stores the data about the test result from the test activation section 246.

[0021] The test activation section 246 is equipped with the functional test activation section 247 and the DC parametric test activation section 248. Based on the data from the circuit tester library 218 stored in a register 242 or memory 244, the test activation section 246 performs a functional test and DC parametric trial to the semiconductor device 250 for inspected, and stores the data of the test result in the test-result storing field of a register 242 or memory 244. In this test activation section 246, two or more driver wave generating sections which create the impression data point impressed to each pin of the semiconductor device 250 for inspected exist. The test-result data stored in a register 242 and memory 244 are incorporated by the circuit tester driver 220 to the direct circuit tester library 218 through the circuit tester bus 230. In addition, the test-result data stored in memory 244 are incorporated to the circuit tester library 218 through a register 242.

[0022] The debugging equipment 100 of drawing 2 simulates actuation of the semiconductor device 250 for inspected while emulating the above-mentioned actuation by the whole semi-conductor testing device 200. Therefore, if the device test program 112 created for the semi-conductor testing devices 200 is performed using the debugging equipment 100 of drawing 2 , actuation of the device test program 112 can investigate whether it is in agreement with what the user meant. Next, the configuration of the debugging equipment 100 concerning the gestalt of this operation is explained.

[0023] The emulator control section 110 shown in drawing 2 is constituted including the device test program 112, an application program 114, the language analysis activation section 116, the circuit tester library 118, and the circuit tester bus emulator 120. This emulator control section 110 is for controlling actuation of the circuit tester emulation section 140, and performs the same actuation fundamentally with the circuit tester control section 210 contained in the semi-conductor testing device 200 shown in drawing 3 .

[0024] The device test program 112 is a program which describes the procedure and approach for what kind of trial is performed to the semiconductor device 250 for inspected using the semi-conductor testing device 200, and is set as the object of debugging with debugging equipment 100. Therefore, the device test program 212 of drawing 3 is transplanted as this device test program 112 as it is, and it is constituted so that same actuation may be performed. The application program 214, the language analysis activation section 216, and the test library 218 of drawing 3 are similarly transplanted as it is about an application program 114, the language analysis activation section 116, and the circuit tester library 118, and it is constituted so that same actuation may be performed. The circuit tester bus emulator 120 drives the virtual circuit tester bus 130 which connects virtually between the emulator control section 110 and the circuit tester emulation sections 140, and controls transmission and reception of the data between the circuit tester library 118 and the circuit tester emulation section 140 through this virtual circuit tester bus 130.

[0025] The circuit tester emulation section 140 realizes actuation of the circuit tester body 240 of drawing 2 by software, and performs the simulation-trial to a virtual device 150 according to directions of the circuit tester library 118 in the emulator control section 110 of operation. The circuit tester emulation section 140 is constituted including the virtual register 142, virtual memory 144, and the virtual test activation section 146. The virtual register 142 stores the data

from the circuit tester library 118. The data stored in this virtual register 142 are sent to the virtual test activation section 146 through direct or virtual memory 144. Moreover, the virtual register 142 and virtual memory 144 have the test-result storing field (not shown) which stores the virtual test-result data outputted from the virtual test activation section 146. The virtual test activation section 146 is equipped with the functional test activation section 147 and the DC parametric test activation section 148. Based on the data from the circuit tester library 118 stored in the virtual register 142, this virtual test activation section 146 outputs a predetermined impression data point to a virtual device 150, performs the functional test by the functional test activation section 147, and DC parametric trial by the DC parametric test activation section 148, and stores that virtual test-result data in the test-result storing field of the virtual register 142 or memory 144. The virtual test-result data stored in the virtual register 142 and virtual memory 144 are outputted to the circuit tester library 118 through the virtual circuit tester bus 130. The test-result analysis judging section 160 carries out comparison examination of the expected value of the test result beforehand expected to be virtual test-result data stored in the virtual register 142, memory 144, or the circuit tester library 118, verifies whether the device test program 112 is operating normally, and displays the result on a user. For example, when the test result which was mistaken with activation of the device test program 112 is obtained, the program line number leading to the mistaken test result etc. is displayed on a monitor (not shown), or is printed out.

[0026] Drawing 1 is drawing showing the outline configuration of the driver wave generating section in the virtual test activation section 147 which creates in false the impression data point impressed to each pin of a virtual device 150. In drawing 1 , the same sign is given to the thing of the same configuration as drawing 4 . The point that the driver wave generating section of drawing 1 differs from the thing of drawing 4 is a point which stores in the event table 50 corresponding to a pattern the event data generated by the frame processor module 30 with pattern data, reads the event data corresponding to pattern data from the event table 50 corresponding to a pattern when the pattern data outputted from the pattern generating module 10 exist in the event table 50 corresponding to a pattern, and was stored in the event table 40.

[0027] In this drawing, the driver wave generating section is constituted including the pattern generating module 10, the pattern judging module 20, the frame processor module 30, the event table 40, and the event table 50 corresponding to a pattern.

[0028] The pattern generating module 10 generates the pattern data of two or more bit configuration according to the pattern program 11, and outputs them to the frame processor module 30. For example, the pattern data of two or more bit configuration outputted from the pattern generating module 10 consist of data of two or more bit configuration with which the pin data for specifying the pin by which a wave is impressed further were added to the data of 5 bit patterns which consist of combination of the data of the triplet configuration which the usual pattern generator generates, the timing set of 4 bit patterns for choosing 16 kinds of timing edge, and the mode data about pattern generating mode.

[0029] The pattern judging module 20 outputs the pattern data to the frame processor module 30, when it judges whether the event data corresponding to the pattern data of two or more bit configuration outputted from the pattern generating module 10 exist in the event table 50 corresponding to a pattern and is judged with it not existing, and when judged with existing, it outputs the pattern data to the event table 50 corresponding to a pattern. In addition, a flag may be formed for every pattern data, by whether "1" is set to the flag, the pattern judging module 20 may be made to judge whether the event data corresponding to pattern data exist in the event table 50 corresponding to a pattern, and direct access may be carried out to the event table 50 corresponding to a pattern, and it may judge it.

[0030] The frame processor module 30 is constituted including the wave memory 31 and the timing memory 32, generates the event data based on the pattern data outputted from the pattern judging module 20, and outputs them to the event table 40. The wave memory 31 generates the data about a wave-like event based on the pattern data outputted from the pattern judging module 10. The data about this wave-like event are data in which the data in which a wave-like standup is shown, or wave-like falling is shown. The timing memory 32 assigns

the timing data in which the generating timing of the event is shown to the data about the wave-like event outputted from the wave memory 31 based on the pattern data outputted from the pattern judging module 20, generates the event data which consist of combination of the data and timing data about an event, and outputs it to the event table 40.
[0031] Sequential storing of the event data read from the event data or the event table 50 corresponding to a pattern outputted from the frame processor module 30 to the event table 40 is carried out as a data point corresponding to each pin. Sequential storing of the same data as the event data stored in the event table 40 is carried out by making into the address the pattern data outputted to the event table 50 corresponding to a pattern from the pattern judging module 20. The event data stored in the event table 50 corresponding to a pattern are read one by one considering the pattern data outputted from the pattern judging module 20 as the address, are outputted to the event table 40 as a data point, and are stored there.
[0032] the pattern generating module 10 mentioned above -- a pattern generating means -- the frame processor module 30 -- a frame processor means -- the wave memory 31 -- wave memory -- the timing memory 32 -- timing memory -- the event table 50 corresponding to a pattern corresponds to the event table means corresponding to a pattern, and the pattern judging module 20 corresponds [the event table 40] to an event table means at a pattern judging means, respectively.
[0033] It explains that the driver wave generating section of drawing 1 operates how according to the pattern program 11. In addition, it explains as that by which the pattern data of two or more bit configuration outputted from the pattern generating module 10 are constituted from the impression pattern data and pin data within the pattern program 11, and omits about the relation between a timing set and mode data.
[0034] First, according to the 1st NOP instruction "NOP !01", the pattern generating module 10 outputs the pattern data "10" corresponding to the pin data "1" and impression pattern data "0" of the 1st pin to the pattern judging module 20. In addition, impression pattern data are data of a triplet configuration, and, in the case of 216 pins, pin data are data of 8 bit patterns, but it expresses as mentioned above on [of explanation] expedient. Since the pattern judging module 20 judges with pattern data "10" not existing in the event table 50 corresponding to a pattern, it outputs the pattern data "10" to the frame processor module 30. The wave memory 31 of the frame processor module 30 generates two, the event which shows a wave-like standup, and the event which shows falling, based on the pattern data "10" outputted from the pattern judging module 20. The timing memory 32 of the frame processor module 30 assigns 70ns based on the pattern data "10" outputted from the pattern judging module 20 as generating timing of an event which shows falling for 20ns as generating timing of an event which shows a standup. Consequently, the data point which consists of event data of starting in timing 20ns, and event data of falling in timing 70ns is stored in the event table 40. Similarly, the data point which consists of event data of starting in timing 20ns, and event data of falling in timing 70ns is stored in the event table 50 corresponding to a pattern by making pattern data "10" into the address.
[0035] Next, the pattern generating module 10 outputs the pattern data "21" corresponding to the pin data "2" and impression pattern data "1" of the 2nd pin of the 1st NOP instruction "NOP !01" to the pattern judging module 20. Since the pattern judging module 20 judges with pattern data "21" not existing in the event table 50 corresponding to a pattern, it outputs the pattern data "21" to the frame processor module 30. The wave memory 31 of the frame processor module 30 generates two, the event which shows a wave-like standup based on the pattern data "21" outputted from the pattern judging module 20, and the event which shows falling. The timing memory 32 of the frame processor module 30 assigns 90ns as generating timing of an event which shows falling for 40ns as generating timing of an event which shows a standup based on pattern data "21." Consequently, the data point which consists of event data of starting in timing 40ns, and event data of falling in timing 90ns is stored in the event table 40. Similarly, the data point which consists of event data of starting in timing 40ns, and event data of falling in timing 90ns is stored in the event table 50 corresponding to a pattern by making the pattern data "21" corresponding to the pin data "2" and impression pattern data "1" of the 2nd pin into the address.

[0036] The pattern generating module 10 outputs the pattern data "11" corresponding to the pin data "1" and impression pattern data "1" of the 1st pin of the 2nd NOP instruction "NOP !11" to the pattern judging module 20. Since the pattern judging module 20 judges with pattern data "11" not existing in the event table 50 corresponding to a pattern, it outputs the pattern data "11" to the frame processor module 30. The frame processor module 30 generates the event data based on pattern data "11", and stores it in the event event table 40 and 50 corresponding to a pattern like the above-mentioned.
[0037] Next, the pattern generating module 10 outputs the pattern data "21" corresponding to the pin data "2" and impression pattern data "1" of the 2nd pin of the 2nd NOP instruction "NOP !11" to the pattern judging module 20. Since the pattern judging module 20 judges with pattern data "21" already existing in the event table 50 corresponding to a pattern, it reads the pattern data "21" and outputs it to the event table 50 corresponding to a pattern as the address. From the event table 50 corresponding to the pattern into which pattern data "21" were inputted as the address, the data point which consists of event data of starting in timing 40ns [finishing / storing], and event data of falling in timing 90ns, by the last processing is read, and it is stored in the event table 40. Hereafter, according to the pattern program 11, same processing is performed similarly repeatedly. Thus, since the event data which processing of the frame processor module 30 was omitted and were read from the event table 50 corresponding to a pattern are only stored in the event table 40, processing becomes easy and a rate improves. Moreover, since the rate that processing of the frame processor module 30 is omitted in such a case since the pattern program 11 is constituted by the repeat of the usually same pattern in many cases also becomes large, execution speed will improve by leaps and bounds.
[0038] In addition, this invention is not limited to an above-mentioned operation gestalt, and deformation implementation various by within the limits of the summary of this invention is possible for it. For example, although the functional block diagram has shown the driver wave generating section of drawing 1 , it cannot be overemphasized that you may realize by the software corresponding to this. Moreover, although the gestalt of above-mentioned operation explained the driver wave generating section which creates an impression data point to the example, also when creating a comparison data point, it cannot be overemphasized that it is similarly applicable.
[0039]
[Effect of the Invention] As mentioned above, even when actuation of a semi-conductor testing device is emulated according to this invention, it is effective in the ability to create the data point corresponding to each pin at a high speed.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1] In the program debugging equipment for a semi-conductor trial concerning this invention, it is drawing showing the outline configuration of the driver wave generating section which creates in false the impression data point impressed to each pin of a virtual device.
[Drawing 2] It is drawing showing the whole program debugging equipment configuration for a semi-conductor trial concerning this invention.
[Drawing 3] It is drawing showing the actual semi-conductor testing-device whole configuration.
[Drawing 4] In conventional debugging equipment, it is drawing showing the outline configuration of the driver wave generating section which creates the impression data point according to a test condition.
[Description of Notations]
10 Pattern Generating Module
20 Pattern Judging Module
30 Frame Processor Module
31 Wave Memory
32 Timing Memory
40 Event Table
50 Event Table corresponding to Pattern

[Translation done.]